EPITA – S3-API – Projet
Cahier des charges

David Bouchet Christophe Boullay

1 Cadre

Le projet est à réaliser en groupe de quatre personnes (maximum). Il s'étend de Septembre à Décembre. Le présent document présente les objectifs du projet, les différentes parties attendues ainsi que le planning de réalisation de celles-ci. Ce document présente également les différentes contraintes de réalisation.

2 Le sujet

L'objectif de ce projet est de réaliser un logiciel de type O.C.R. (*Optical Character Recognition*) dont la fonctionnalité principale est d'extraire depuis une image (*bitmap*) le texte contenu dans celle-ci.

Votre application prendra donc une image en entrée dans un format *standard* et produira un texte reconnu. Dans sa version finale, votre application disposera d'une interface graphique permettant de charger l'image, de la visualiser et éventuellement corriger le texte produit et enfin de sauver celui-ci. Votre application aura également un aspect apprentissage, qui pourra être séparé de la partie principale, permettant d'entraîner votre réseau de neurones et de sauver et/ou recharger le résultat de cet apprentissage.

2.1 Description détaillée

Conceptuellement, le flux de traitement de votre OCR sera plus ou moins le suivant :

1. chargement de l'image ;
2. suppression des couleurs (niveau de gris, puis noir et blanc) ;
3. pré-traitement (*optionnel*) ;
4. détection des blocs de textes ;
5. détection des caractères ;
6. reconnaissance des caractères détectés ;
7. reconstruction du texte.

Les phases initiales (chargement, suppression des couleurs et pré-traitement éventuel) permettent de rentrer dans la partie principale de l'OCR. On notera qu'aucune information de couleur n'est nécessaire pour la reconnaissance de caractères et qu'il est même fortement conseillé de travailler en noir et blanc.

2.2 Découpage de l'image

Le découpage de l'image en blocs puis en caractères (en passant éventuellement par un découpage des blocs en lignes) est nécessaire pour envoyer à la reconnaissance les morceaux d'images correspondants logiquement à des caractères.

Il est fortement conseillé de penser cette partie de manière récursive et modulaire. En effet, si vous voulez préserver la mise en page du texte reconnu (paragraphes, passages à la ligne, multicolonnes ...) vous devrez récupérer les informations nécessaires pendant cette partie.

2.3 Reconnaissance de caractères : réseau de neurones

La reconnaissance de caractères est la partie centrale de votre OCR. Cette partie nécessite une phase d'apprentissage pendant laquelle votre réseau de neurones va apprendre à reconnaître les différents caractères.

Un réseau de neurones est un outil permettant d'apprendre une fonction (possiblement non linéaire) à l'aide d'exemples : dans la phase d'apprentissage du réseau vous allez fournir des entrées déjà identifiées (ici des blocs d'images) auxquelles votre réseau répondra. La marge d'erreur de ces réponses sera alors utilisée pour *corriger* les poids internes du réseau. Avec un certain nombre d'exemples, votre réseau finira, dans la majorité des cas, par fournir les bonnes réponses. La sortie usuelle pour ce genre de problème est une probabilité par symbole du jeu de caractères. Le symbole avec la plus forte probabilité sera retenu comme le symbole reconnu.

Pour implémenter votre réseau de neurones, voici quelques termes à rechercher : ***apprentissage supervisé***, ***perceptron multicouche***, ***rétropropagation***, ***descente de gradient***.

Le site `http://neuralnetworksanddeeplearning.com/` (en anglais) fournit beaucoup d'informations théoriques et pratiques qui pourraient vous être utiles (notamment pour la partie *paramétrage* de votre réseau située dans le chapitre 3. du site)

Il existe de nombreuses formes de réseau de neurones, mais pour votre projet vous n'avez probablement pas besoin de plus d'une couche cachée. Par contre, il peut être judicieux de remplacer la fonction d'activation de la dernière couche par un softmax plutôt que la fonction sigmoïde habituelle. Vous trouverez toutes ces informations dans le lien fourni plus haut.

2.4 Partie optionnelle : pré-traitement

L'objectif de cette partie est d'améliorer la qualité des images traitées. Cette partie n'est pas *nécessaire* dans l'ensemble de la chaîne de traitements, mais elle permettra d'accepter en entrée des documents directement issus d'un scanner ou d'une photo. Voici quelques exemples de traitements pertinents pour votre projet :

— redressement de l'image (détection d'angle de rotation et rotation de l'image),
— élimination des bruits parasites (grain de l'image, tâches...),
— renforcement des contrastes...

3 Les contraintes du projet

Votre projet sera réalisé en C et devra compiler et fonctionner dans l'environnement de travail qui vous est fourni (Linux). Sauf cas particulier, seuls les outils déjà installés pourrons être utilisés. Vous devrez respecter les points suivants :

— La version du standard C utilisée est **C99**.
— Vous pourrez utiliser **gcc** ou `clang` pour compiler votre code.
— Votre code doit compiler, sans warning avec au moins les options suivantes : `-Wall -Wextra -std=c99`
— Votre projet sera livré avec un fichier `Makefile` fournissant au moins les règles `all` et `clean`.
— Votre projet fournira un fichier `README` décrivant de manière concise comment utiliser votre application.
— Votre projet fournira un fichier `AUTHORS` contenant les logins des membres de votre groupe au format décrit plus loin.
— Tous les identifiants utilisés dans votre code ainsi que les commentaires devront être en anglais.
— Dans votre code, les lignes ne devront pas dépasser 80 colonnes.
— Il ne devra pas y avoir d'espaces en fin de ligne.

Le fichier `README` doit être au format texte et pouvoir être lu correctement dans un terminal avec une commande comme `cat` ou `less`. Il pourra éventuellement être au format *markdown*. Le fichier `AUTHORS` devra contenir une ligne par membre du groupe sous la forme : `* login (NOM Prenom)`

3.1 Contrôle de versions et rendus

Pour le projet vous allez disposer d'un dépôt `git` utilisable par tous les membres de votre groupe. Il est fortement conseiller d'utiliser les fonctionnalités de gestion de version de `git`.

Pour chaque soutenance un rendu de code vous sera demandé avant la soutenance.

Ce rendu de code s'effectuera à l'aide de `git` : les examinateurs effectueront un *clone* de votre dépôt sur la branche principale (`master`) à la date et à l'heure prévues pour le rendu. Seuls les fichiers présents sur votre dépôt au moment du *clone* seront utilisés pour montrer votre projet pendant la soutenance.

Votre dépôt devra contenir :

— Le fichier `AUTHORS`
— Le fichier `README`
— Votre code source et son fichier `Makefile`

Le dépôt ne devra pas contenir : de fichiers exécutables, de code source extérieur (bibliothèques, autres projets…), de fichiers binaires autres que des images de démonstration nécessaires à la soutenance et de rapport (sous quelque forme que ce soit) …

4 Planning et livrables

Les dates clefs du projet sont les suivantes :

21 septembre 2018 : constitution des groupes

22 octobre 2018 : première soutenance (éventuellement en fin de journées & soirées de la semaine)

3 décembre 2018 : soutenance finale (éventuellement en fin de journées & soirées de la semaine)

La première soutenance est une soutenance de suivie, l'objectif est de montrer votre état d'avancement, étant donné que vous faites tous le même sujet, vous n'avez pas besoin de présenter celui-ci, ni de paraphraser le présent document.

Pour cette soutenance, vous devez prévoir une introduction et une conclusion, ainsi qu'une démonstration des fonctionnalités attendues, vous devrez également expliquer le fonctionnement de votre réseau de neurones. Attention : là encore, il ne vous est pas demandé de faire un exposé sur les réseaux de neurones, mais bien de décrire le fonctionnement de celui que vous avez implémenté.

Pour la soutenance finale, vous devez faire la démonstration d'un programme fini, il est important que celui-ci soit homogène et permette d'effectuer la chaîne de traitements attendus. Il est plus important d'avoir un vrai programme (et pas une collection de bout de code) même si celui-ci ne fonctionne que sur un jeux réduit d'exemple.

4.1 Livrables pour chaque soutenance

En début de semaine de soutenance, vous devrez fournir (comme présenté dans la section 3.1) le code de votre projet qui sera utilisé pendant la soutenance. Lors de la soutenance, vous devrez fournir :

— Un rapport présentant votre travail (minimum 25 pages pour la première soutenance et 40 pour la soutenance finale) : présentation du groupe, répartition des charges, état d'avancement…
— Un plan de soutenance.

4.2 Découpage du projet et livrables techniques

Votre projet se découpe en plusieurs éléments. Un élément est considéré comme étant réalisé si votre programme est capable de réaliser la tâche attendue sur quelques exemples non triviaux. Bien évidement, l'évaluation complète de l'élément dépendra de la qualité et de la robustesse face à des tests avancés.

- **Éléments devant être réalisés obligatoirement pour la première soutenance :**
  - chargement des images et suppression des couleurs ;
  - détection et découpage en blocs, lignes et caractères ;
  - *proof of concept* de votre réseau de neurones : au minimum un mini réseau capable d'apprendre la fonction *XOR*.

- **Éléments devant être commencés pour la première soutenance :**
  - sauvegarde et chargement des poids du réseau de neurones ;
  - jeu d'images pour l'apprentissage ;
  - manipulation de fichiers pour la sauvegarde des résultats

- **Éléments devant obligatoirement être réalisés pour la soutenance finale :**
  - tous les éléments obligatoires de la première soutenance ;
  - Le réseau de neurones complet et fonctionnel (apprentissage et reconnaissance) ;
  - reconstruction du texte et sauvegarde ;
  - Une interface utilisateur permettant d'utiliser ces éléments.

- **Éléments supplémentaires (évalués uniquement si la partie obligatoire est faite) :**
  - pré-traitement
  - intégration d'un correcteur orthographique ;
  - bonus : gestion du multicolonne, mise en page, images dans le texte…

5 Conseils

Ce projet est conçu pour être réalisé en équipe de quatre personnes sur une période totale de trois mois. Certains éléments, comme le réseau de neurones, nécessitent des phases d'ajustement et parfois également des phases de calcul pouvant prendre du temps. **Il est donc important de commencer à travailler dès la constitution de votre groupe, sous peine de ne pas disposer d'assez de temps pour finaliser votre projet.**

Il peut être également judicieux de concevoir des tests indépendants pour chaque partie. Par exemple, le réseau de neurones devra recevoir en entrée des images de caractères déjà découpées et redimensionnées. Afin de tester et d'entraîner votre réseau de neurones, il serait pertinent de disposer d'un jeux de tests où les images sont déjà des caractères seuls au bon format. Ainsi, même si votre découpage de l'image n'est pas prêt, vous pourrez faire avancer cette partie.

Lors de la première soutenance, nous vous demanderons de faire la démonstration des parties attendues. Pensez à préparer votre projet pour faire la démonstration de ces parties de manière simple et efficace (visualisation intermédiaire, exécutions indépendantes …).

Il y a quelques écueils en C qui pourraient vous faire perdre beaucoup de temps. Vous aurez besoin de porter une attention particulière à de nombreux détails techniques. Prenez de bonnes habitudes pour éviter de perdre du temps. Notamment, pour tout vos problèmes de mémoire, utilisez un outil comme `valgrind` ou l'option `-fanalizer=address` de `gcc/clang`. Notez également, que la meilleure représentation des tableaux à plusieurs dimensions (notamment les matrices) est la représentation à une seul dimension en calculant les indexes *à la main.*